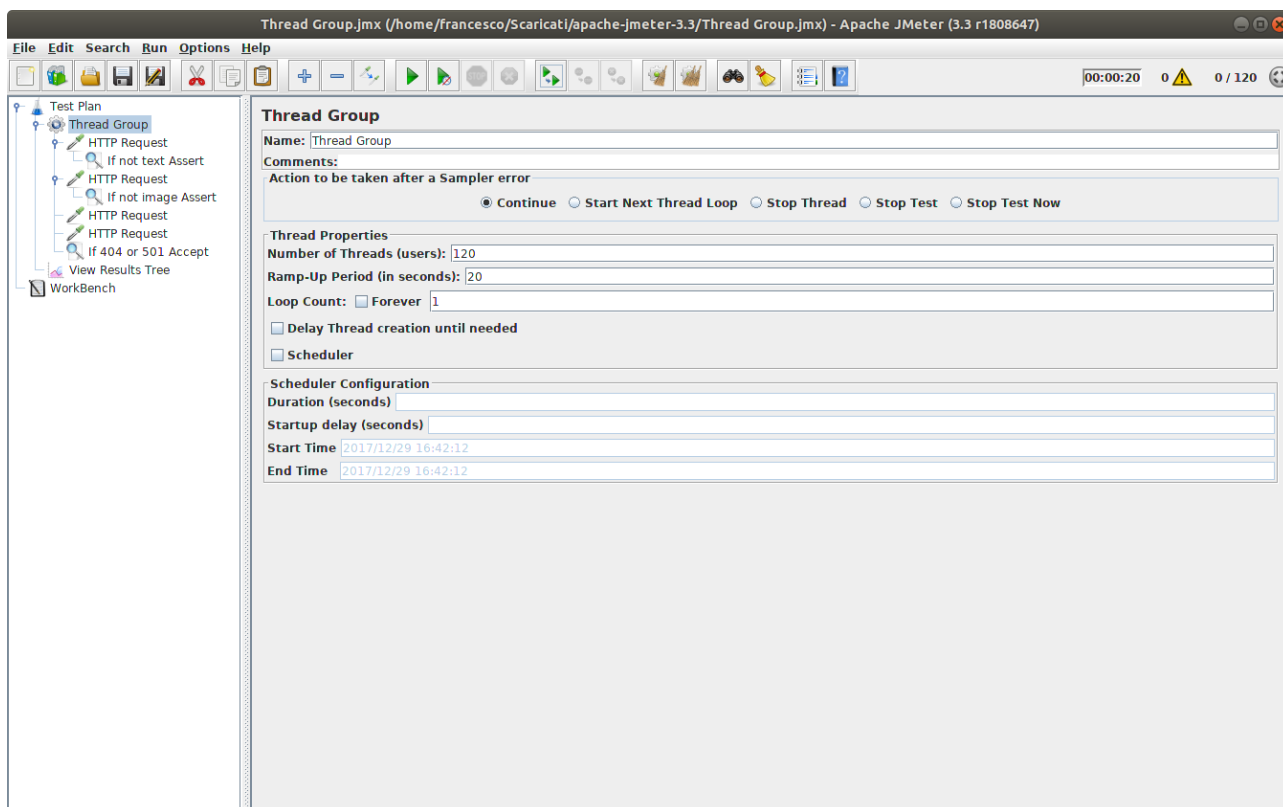


# Stress Test

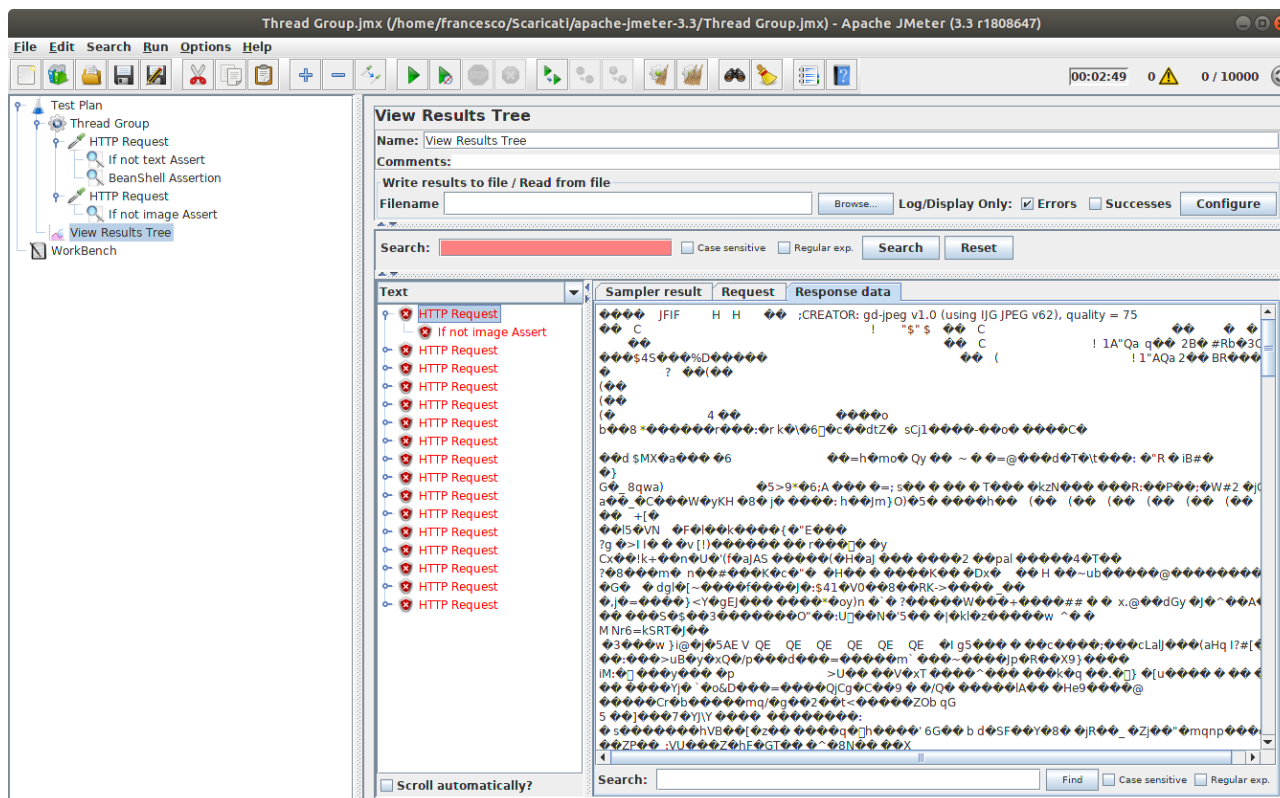
Per verificare la mutua esclusione tra i thread di risposta della versione 2.2 e la giusta implementazione del pipelining, è necessario uno stress test di richieste HTTP. Questo è stato effettuato tramite l'apposita applicazione JMeter.



*Interfaccia GUI JMeter*

In questo caso ho utilizzato esclusivamente richieste HTTP, tuttavia JMeter implementa l'utilizzo di molti altri protocolli. Attraverso la semplice interfaccia è possibile configurare molti parametri e strumenti utili come asserzioni o listener che controllano i risultati ottenuti.

Per i miei vari stress test ho impostato che ogni thread faccia 4 differenti richieste HTTP: la prima della pagina index.html, la seconda di uncadunca.jpg, la terza di 404\_Not\_found.html e infine 501\_method\_not\_implemented.html. In questa immagine il numero di thread è impostato a 120 ogni 20 secondi, quindi 480 richieste HTTP ogni 20 secondi, ovvero 24 al secondo. Tramite svariate prove, questo è il numero che riesce a gestirne in un secondo, se provo ad aumentarlo infatti, il test impiega più di 20 secondi per essere terminato.



### *Risultato Race Condition mal gestita*

In questa immagine è invece interessante notare cosa accade nel caso in cui non sia gestita la race condition durante l'ottenimento del mime. Attraverso qualche asserzione è possibile filtrare i pacchetti con un risultato sbagliato, ovvero se richiedo un testo ma ricevo un binario e viceversa. Il risultato è l'errata interpretazione del payload del pacchetto come mostrato in figura. La possibilità che ciò avvenga è molto bassa e senza uno strumento apposito sarebbe stato impossibile individuarlo, infatti c'è stato bisogno di una mole notevole di richieste al secondo per generare gli errori.